



TITLE:

xfy におけるContent MathML の編集プラグインの開発と応用(数式処理研究の新たな発展)

AUTHOR(S):

河田, 貴幸; 中西, 智美; 甲斐, 博; 田村, 恭士

CITATION:

河田, 貴幸 ...[et al]. xfy におけるContent MathML の編集プラグインの開発と応用(数式処理研究の新たな発展). 数理解析研究所講究録 2007, 1572: 39-49

ISSUE DATE:

2007-11

URL:

<http://hdl.handle.net/2433/81309>

RIGHT:

xfy における Content MathML の編集プラグインの開発と 応用

河田 貴幸

TAKAYUKI KAWATA *

中西 智美

TOMOMI NAKANISHI †

甲斐 博

HIROSHI KAI ‡

愛媛大学大学院理工学研究科

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING, EHIME UNIVERSITY

田村 恭士

YASUSHI TAMURA §

株式会社ジャストシステム

JUSTSYSTEMS CORPORATION

1 まえがき

近年, インターネットの普及に伴い, XML を利用した様々な技術が発展している. XML は W3C に勧告を受けた文書やデータの意味と構造を記述するためのマークアップ言語であり, ユーザが独自のタグを定義できるため非常に拡張性が高い. また, HTML と互換性を持つ XHTML, 図形描画用の SVG, 数学記述用の MathML 等, 様々な分野における標準化された XML 文書が考案されている. 今後, 文書の再利用や多様なソフトウェアの間のデータ交換などのために, XML による文書表現はますます重要になってくると考えられる.

このような中, あらゆる XML ボキャブラリを組み合わせた XML 複合文書を WYSIWYG で作成・編集することを目的としたソフトウェアである xfy[2] が発表された. xfy は Java により開発されており, プラットフォームに依存しないため移植性が高く, またプラグインや XVCD を開発をすることで XML 編集機能の拡張が可能である.

現在 xfy が利用される場面は, xfy 単体で用いる一般の文書編集のほかに, RDB などのソフトウェアシステムのフロントエンドやブログなどの Web 上の文書編集など多くの応用が考えられる. 今後は, e-learning のコンテンツ作成や LMS 管理など教育面での利用をはじめ, XML を用いた場面でのさまざまな利用が期待できる.

本研究では xfy の数学文書処理に関する研究を目的とし, xfy の数式処理インタフェースについて考察する. 数式処理の計算や結果を xfy で容易に利用できるようになると, 技術文書編集や教育コンテンツ編集などを行う際に役立つ.

*kwttkyk@hpc.cs.ehime-u.ac.jp

†nakanishi@hpc.cs.ehime-u.ac.jp

‡kai@cs.ehime-u.ac.jp

§Yasushi.Tamura@justsystem.co.jp

Maple や Mathematica などの数式処理システムでは、現在 MathML をサポートしており、MathML により数式の入出力が行えるようになっている。しかし XML インタフェースを持つ数式処理システムはまだ一部であり、数式処理システムを統一的に扱うことのできるフロントエンドはまだない。一方、xfy では MathML Presentation Markup が編集可能であるが、これは数式表記を主に考えた仕様であるため計算には向いていない。

本論では、数式の意味を表現する MathML Content Markup のエディタをプラグインとして実装することを考える。また、数式処理機能を利用する方法として、XML との親和性が高い数学 Web サービスの利用を検討する。これにより、様々な数式処理システムを利用し、計算やその結果をシームレスに文書編集に利用できる数学文書編集環境が構築できる。

2 xfy

xfy は、XHTML や SVG 等のあらゆる XML ボキャブラリが組み合わされた XML 複合文書の編集を目的としてジャストシステム社で設計・開発された統合 XML アプリケーション開発／実行プラットフォームである。また、そのユーザ・インタフェースに関する特徴として、XML 文書の作成・編集を WYSIWYG で行うことができる。また、xfy では、xfy コンポーネントによる機能拡張を可能にするプラグガブル・アーキテクチャを採用しており、あらゆる状況において XML 文書の最適な編集環境を実現する。

xfy コンポーネントとは、XML 文書の編集、XML 部分木に対する加工や計算、ファイル入出力等の機能をそれぞれ個別に実装したものである。また、xfy コンポーネントは主に Java 言語で開発され、それらを xfy プラットフォーム上で利用するために JAR 形式でアーカイブされる。xfy には、予め基本的な機能を提供する xfy コンポーネントが用意されており、XML 文書の閲覧・編集ができる。また、新規に xfy コンポーネントを開発することにより、既存の xfy コンポーネントと組み合わせて利用することも可能である。

2.1 XML 複合文書の処理

xfy における XML 複合文書の処理の流れを図 1 に示す。この例では XHTML と SVG が XML 複合文書に含まれている。

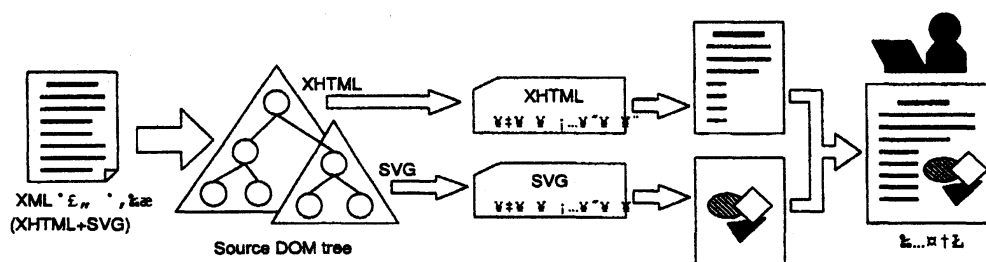


図 1: xfy における XML 複合文書の処理の流れ

Step1 XML 複合文書から DOM ツリー (Source DOM ツリー) を生成する。

Step2 それぞれの名前空間により DOM ツリーを区画化し、対応するボキャブラリコンポーネントによりそれぞれの処理を行う。

Step3 Step2. で行われた処理結果を統合し、XML 複合文書として表示する。

xfy では、XML 複合文書のシームレスな処理を目的としているため、文書構造に関する多くの情報を取得し、利用する機会が多く、また、必要に応じて要素の並び替えや、文書内の任意位置に要素の移動、追加、削除を行わなくてはならない。そのため、xfy では XML 文書の汎用的な処理のための API である Document Object Model (DOM) を利用している。

2.2 XVCD

XVCD は XSLT を拡張したスクリプト言語であり、XML 文書の変換及び編集が行える。XVCD による XML 文書の変換は次のように行われる。xfy は XML 文書に対する DOM ツリー (Source DOM tree) を保持しており、XVCD によって XML 文書を変換することで新たな DOM ツリー (Destination DOM tree) を得る。Destination DOM tree の表示は xfy コンポーネントにより行われ、ユーザは XML 文書の閲覧が可能となる。またユーザが Destination DOM tree に対する編集を行うことで、その編集による変更は直ちに Source DOM tree に反映される。XVCD は XML 文書をユーザ独自の表示と編集のためのプラットフォームである。

3 MathML とその編集機能について

あらゆる情報交換が電子的に行われている現在において、特に技術的な情報に関し、それらを検索、再利用されることが求められている。しかし、現在の Web 上における数式の表現は、主に画像を用いるため、画像に含まれる数学的な情報を検索、あるいは他のアプリケーションにより再利用することができない。そこで、そのような問題を解決するために数式記述用の XML として MathML が W3C に勧告された。

MathML には、数式の表記構造を記述する Presentation Markup と、数学的な意味を記述する Content Markup がある。MathML を記述するには、Presentation Markup と Content Markup のどちらか一方のみでの記述、もしくは両方の記法を混在・並列させて記述することもできる。また、MathML は単独でも使用可能であるが、名前空間を用いることで XHTML 等の他の XML 文書に埋め込むことも可能である。

Content Markup は、数式の数学的な意味を記述するものであり、約 150 種類の要素が用意されている。また、それらの要素は「Token Elements」、「Basic Content Elements」、「Arithmetic, Algebra and Logic」、「Relation」、「Calculus and Vector Calculus」、「Theory of Sets」、「Sequences and Series」、「Elementary classical functions」、「Statistics」、「Linear Algebra」、「Semantic Mapping Elements」、「Constant and Symbol Elements」のいずれかに分類される。

3.1 Content Markup 編集上の注意点

Content Markup はアプリケーションが数式の計算などに関わる処理を容易にするために、数式の数学的な意味を備えた構造として記述される。Content Markup を編集する際には、次のことに注意すべきである。

注意 1 編集後もその文書構造が数学的な意味を備えている

注意 2 数式が複数の表記を持つことがある

例として、図 2 に示す数式 $x - y$ を $x - y - z$ に編集した際の編集結果を図 3 に示す。但し、math 要素は省略している。

注意 1 を解決するための一つの方法は、テンプレートによる編集方法である。テンプレートそのものは XML 文書の部分木として表現することができる。前述の例をテンプレートを用いて編集する場合、図 4 に

```

<apply>
  <minus/>
  <ci> x </ci>
  <ci> y </ci>
</apply>

```

図 2: 数式 “ $x - y$ ” の XML 構造

```

<apply>
  <minus/>
  <apply>
    <minus/>
    <ci> x </ci>
    <ci> y </ci>
  </apply>
  <ci> z </ci>
</apply>

```

図 3: 編集結果

```

<apply>
  <minus/>
  <csymbol></csymbol>
  <csymbol></csymbol>
</apply>

```

図 4: 二項減算テンプレート

示すテンプレートを用いる。ここで、テンプレートに含まれる csymbol 要素は minus 要素が適用される被演算子要素であり、また新たな入力枠としての役割を持つものとする。さらに、この csymbol 要素に対して数や変数が入力されることで cn 要素や ci 要素に変化することを想定している。削除を行う場合は、数式の意味を失わないためにテンプレート単位での削除を行う。但し、編集の容易さのため、図 2 の状態から y のみを削除した場合は、図 5 の状態になるべきである。

```

<apply>
  <minus/>
  <ci> x </ci>
  <csymbol></csymbol>
</apply>

```

図 5: csymbol 要素による補完

また、注意 2 に関しては、例えば、数式 $x + y \times z$ を Content Markup で記述しそれを表記するためには、 $x + y \times z$ という書き方と $x + (y \times z)$ という書き方が存在する。また、優先度が存在しないような場合には、図 6、図 7 に示す 2 種類の記述方法があってもよい。ここで、それぞれの Content Markup をその構造が持つ演算順序等の数学的意味を考慮した上で数式を表現すると、“ $(x + y) \times z$ ” (図 6)、“ $x + (y \times z)$ ” (図 7) となり、それぞれ数学的意味の異なる数式である。よって、図 6 の場合において、少なくとも “ $x + y$ ” に対して括弧を付加しなければ Content Markup の構造が表す数学的意味とエディタに表示される数式の持つ数学的意味が異なる。

そのため、編集中の Content Markup の構造とエディタに表示される数式の持つ数学的意味を対応させるため必要に応じて括弧を数式に加える必要がある。このことは、WYSIWYG 環境で Content Markup 編

```

<apply>
  <times/>
  <apply>
    <plus/>
    <ci> x </ci>
    <ci> y </ci>
  </apply>
  <ci> z </ci>
</apply>

```

図 6: 数式 “ $x + y \times z$ ” の記述例 1

```

<apply>
  <plus/>
  <ci> x </ci>
  <apply>
    <times/>
    <ci> y </ci>
    <ci> z </ci>
  </apply>
</apply>

```

図 7: 数式 “ $x + y \times z$ ” の記述例 2

集する上でも必要なことの1つである。Content Markup において括弧を含む数式の表示を行う方法として次の2つの方法が考えられる。

方法 1 括弧の表示に関する情報をエディタの内部データとして保持させる。

方法 2 括弧を表示用の Presentation Markup 要素である `mfenced` 要素を XML 文書内に埋め込む。

方法 1 は括弧に関する情報を文書を保存できないため方法 2 についてのみ考える。方法 2 を図 6 に適用すると、図 8 に示す文書構造となる。

```

<apply>
  <times/>
  <mfenced open="(" close=")" >
    <apply>
      <plus/>
      <ci> x </ci>
      <ci> y </ci>
    </apply>
  </mfenced>
  <ci> z </ci>
</apply>

```

図 8: `mfenced` 要素を用いた数式 “ $(x + y) \times z$ ” の記述例

Content Markup を編集する既存のエディタには Formulator MathML Weaver[3], Integre MathML Equation Editor[4] などがあるが、これらの注意点の解決が不十分なことがある。

Formulator MathML Weaver の特徴は、テンプレートによる編集やキーボード入力を採用しており、キーボード入力による数式テンプレートの入力も可能である。また、数多くの MathML 要素に対応しており、Presentation Markup との混合記述も可能である。しかし、問題点としては、編集直後における Content Markup の構造の変更がエディタに表示される数式に自動的にフィードバックされない場合がある。例として数式 “ $x \times (y + z)$ ” を意味する Content Markup を作成する場合を考える。この Content Markup を編集するには乗算と加算のテンプレートを順に入力するが、表示される数式は $x \times y + z$ であり、演算順序を意味するための括弧は自動で付加されない。

Integre MathML Equation Editor の特徴は、多項式などの数式編集をオーバーレイを利用することで、テキストエディタのような直接編集を実現している。また、テンプレートによる編集も可能である。基本

的に入力された数式を構文解析することで Content Markup, もしくは Presentation Markup を出力する. 問題点としては, minus 要素が多項演算子として扱われる場合があり, 図 9 のような数学的構造が曖昧な Content Markup ができることである.

```
<apply>
  <minus/ >
  <ci> x </ci>
  <ci> y </ci>
  <ci> z </ci>
</apply>
```

図 9: 数学的構造が曖昧な Content Markup

4 ContentMarkupEditor プラグイン

本節では xfy 上で Content Markup を作成・編集することを目的として開発した xfy コンポーネントについて述べる. Content Markup の編集を容易にするためには, GUI による入力インタフェースを備える必要がある. Padovani らは数式エディタが備えるべき入力インタフェースを既存の Presentation Markup エディタ等の様々な挙動を調査し, 望まれる機能 [5] を次のようにまとめている.

入力スロットの表示 入力スロット (以下, スロット) は, ユーザに対してデータが未入力である部分を明示的に表示することで, その位置へ値の入力を促すためのものである. 入力スロットの表示は “?” や “■” 等のように表現は様々である.

編集位置の表示とその移動 数や変数などを挿入する位置を示すためにキャレットが用いられる. 一般にこのキャレットは点滅する垂直の棒として表現される. また, キャレットは編集位置を表すものである所以移動可能である必要があり, その望ましい挙動として方向キーによる上下左右への幾何的な移動ができること, また移動が対称であることである. 対称な動きとは ‘→’ キーによる移動後, ‘←’ キーにより以前の移動経路と逆順にたどることができる動きであり, 上下移動についても同様である.

編集領域の選択 現在選択されている領域をフォーカスがある領域があると呼び, その領域が編集可能であることを表す. 領域には入力スロットを含む. またフォーカスの表示には, ある領域を実線や破線の矩形で囲んだり, あるいは領域内の文字の色や背景色を反転させるなどの手法が用いられる. フォーカスの選択はマウスによるポインティングやドラッグ操作, あるいは Shift キー+方向キーによるキャレットの移動等で行われる.

カット (コピー) & ペースト フォーカスのある領域を切り取り, あるいはコピーして別の編集位置に貼り付ける編集操作である. この編集操作により, 既存の数式の記述情報を再利用することが容易となる.

テンプレートとオーバーレイ Content Markup において数式の持つ構造を入力する方法としてテンプレートやオーバーレイが利用できる. これらの入力方法は, 数式の構造を表したアイコン等を選択することでキャレット位置に入力できる. テンプレートとオーバーレイは類

似する入力方法であるが、前者の方法は先に数式の構造を与えた後に残りの入力スロットに対して数式を入力する方法であり、一方、後者の方法では既存の数式に対して新たに別の数式の構造を与えることができる入力方法である。

数式の直接編集 数式を容易に編集するためには、テキストエディタでのキーボード入力による直接編集が望ましい。多項式のような頻繁に利用される数式を直接編集できることは、編集時におけるキーボードとマウスの交互利用回数の軽減に繋がるため、ユーザの編集に対する煩わしさをも軽減できると考えられる。

これらの特徴を備えた Content Markup 編集用の xfy コンポーネントを「ContentMarkupEditor プラグイン (以下、本エディタ)」と呼ぶことにし、以下では本エディタの持つ入力インターフェースや編集機能について述べる。

4.1 入力スロット

本エディタにおける入力スロットの表記は “[?]” であり、math 要素が子要素を持たない場合やトークン要素が値を持たない場合等に表示される。図 10 は実際に本エディタ上で表示される入力スロットの様子である。

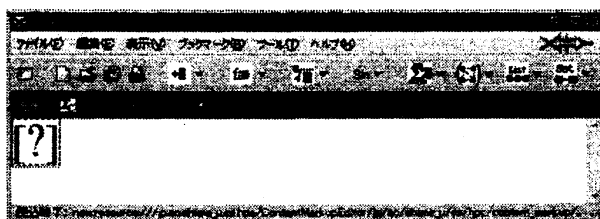


図 10: 入力スロット

4.2 キャレットの表示と幾何的な移動

本エディタにおけるcaretは青色の縦棒として表現している。但し、現実装段階において点滅はしない。また、caretのTabキーや方向キーによる移動について、‘←’キー、‘→’キーによる左右対称な移動を達成している。図 11 は‘→’キーのみによる本エディタでのcaretの表示、及びその移動の様子を示す。

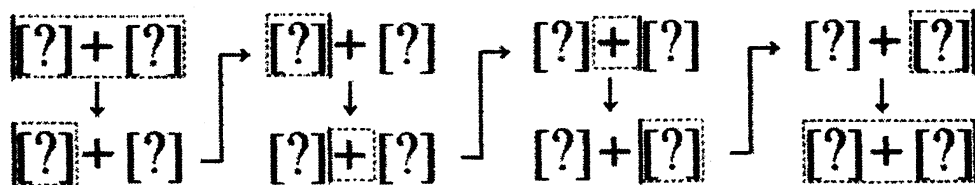


図 11: キャレットの動き

4.3 入力フォーカス

本エディタにおける入力フォーカスによる編集状態を図 12 に示し、それぞれの状態について以下に述べる。

- FOCUS_SELECTED 状態

破線の矩形で囲まれたノードが選択された状態であり、入力が可能な状態。また、この状態にあり、かつキャレットが左側（右側）にある場合に DELETE (BACKSPACE) キーが入力されると、FOCUS_DELETABLE 状態に遷移する。

- FOCUS_CONTEXT 状態

背景色が色付けされたノードのテキストノード値を編集するための状態。ci 要素や cn 要素が持つテキストノード値の編集用の状態。

- FOCUS_DELETABLE 状態

文字色や背景色が反転したノード及びその子ノードの削除が可能な状態。

- FOCUS_OUT 状態

ノードが編集対象でない状態。





	FOCUS_SELECTED状態
	FOCUS_CONTEXT状態
	FOCUS_DELETABLE状態
	FOCUS_OUT状態

図 12: フォーカスによる編集状態

4.4 入力例

本エディタにおけるテンプレート入力の様子を図 13、及び図 14 に示す。ここでは総和を入力する例を用いている。本エディタでは Integre MathML Euation Editor と同様に、キーボード入力からのオーバーレイの入力ができ、テキストエディタによる自然な数式編集に近い編集が行える。例えば、本エディタで数式 $x + y - 1$ を入力する場合、キー入力は “x, +, y, -, 1” の順に入力すればよく、中置記法を意識した数式の編集が行える。

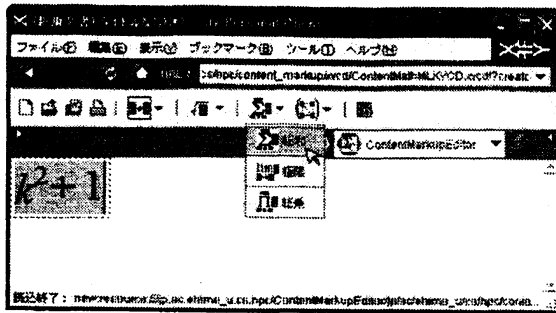


図 13: テンプレートの選択

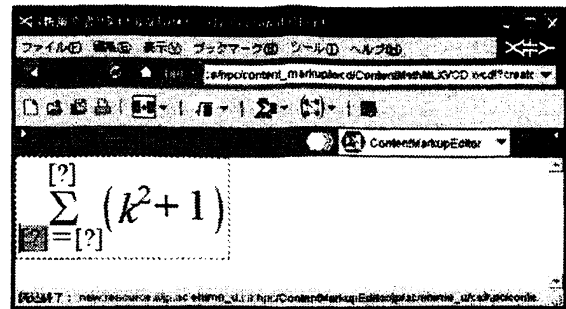


図 14: テンプレート入力

5 xfy 上での数式処理

数式処理システムのインタフェースは様々であり、その計算機能を Web サービスとして利用するための仕組みが数学 Web サービスとして研究されてきた [6, 7]. 数学 Web サービスでは、サービスの内容やインタフェースを WSDL に記述し、数式処理機能を統一的に http 上の SOAP プロトコルで Web 上から利用できる. 本論では、Lupin 数学 Web サービス [6] を xfy から利用する方法について検討する. すなわち、xfy は数式の編集や数学 Web サービスの実行コマンドを扱うためのフロントエンドとしての働きをする.

xfy 上で Web サービスを用いて数式処理を利用するためには、そのためのクライアントを作成する必要がある. 我々は SOAP 通信を行うためにプラグインを開発し、Web サービスを利用するためにプライベート XML を定義した.

クライアントの構成図を図 15 に示す. プライベート XML は、WSDL, SOAP, MathML 等を含んだ XML 複合文書である. プライベート XML は XVCD を用いて xfy 上で編集できるようになる. 計算の際に

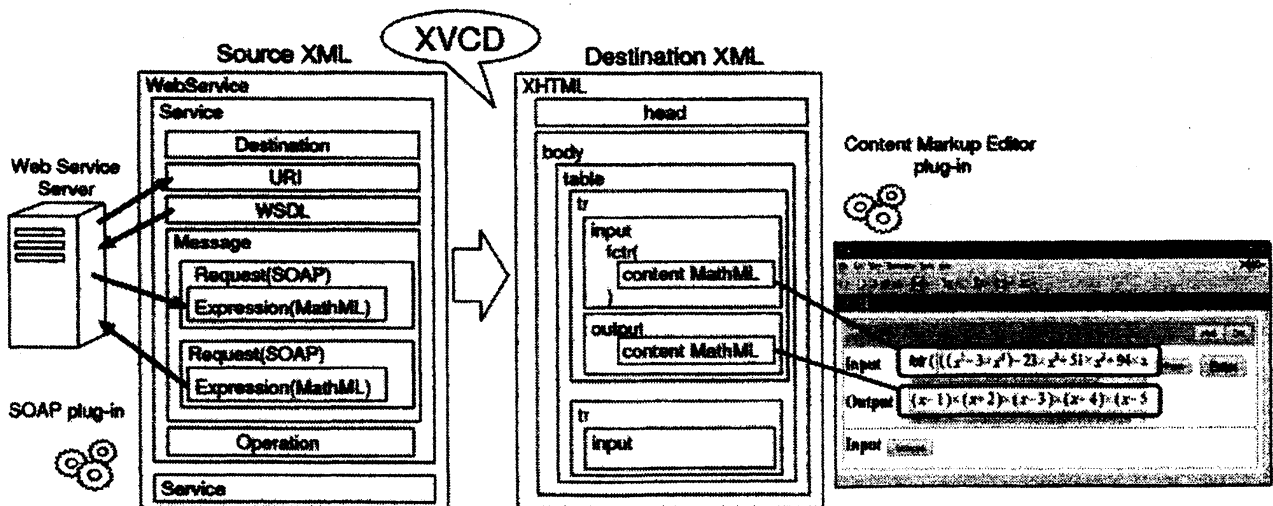


図 15: Web サービスクライアントのアーキテクチャ

は、まず Web サービスのインタフェースを記述した WSDL を選択する. また、本研究で作成した Content Markup プラグインを用いて必要な数式を入力する. 入力がある SOAP プラグインにより Web サービスに渡されることにより Web サービスが実行される. 結果は Content Markup として返されるので、xfy はプラグインを用いてそれを表示する. 以上のような手順で計算が可能になる. これらの全ての操作は同じワーク

スペース上で行われている。

Lupin における因数分解の数学 Web サービスへのアクセスの例を示す。WSDL ファイル (AsirFctrService.wsdl) はクライアントにロードされ, xfy はクライアントを生成する (図 17)。数式処理システム Risa/Asir の fctr コマンドが入力 $x^5 + x^4 - 2y^2x^3 - 2y^2x^2 + y^4x + y^4$ に対して数学 Web サービスにおいて実行され (図 18), 結果 $(x+1)(x+y)^2(x-y)^2$ が MathML プラグインを用いてワークスペース上に表示される (図 19)。

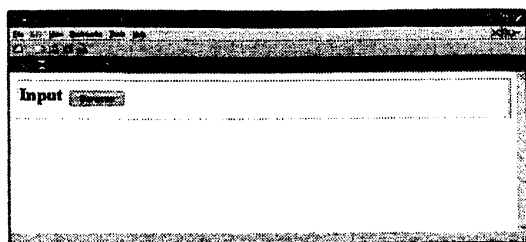


図 16: 実行開始画面

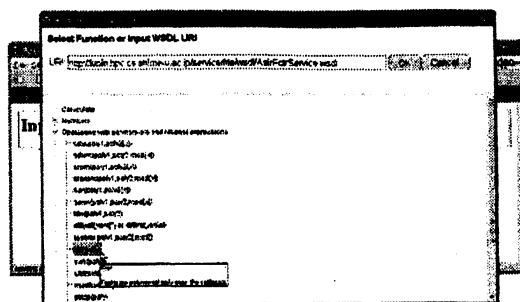


図 17: Web サービスの WSDL を選択

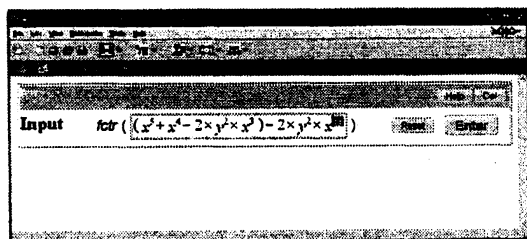


図 18: 本エディタによる Content Markup の編集

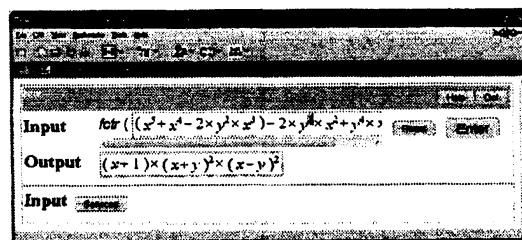


図 19: Web サービスの出力結果

xfy は XML 複合文書を編集するソフトウェアであるので, ブログ文書 (図 20) や e-learning コンテンツ (図 21) 等といった他のアプリケーションにおいて容易に数式処理の計算結果を用いることができるようになる。

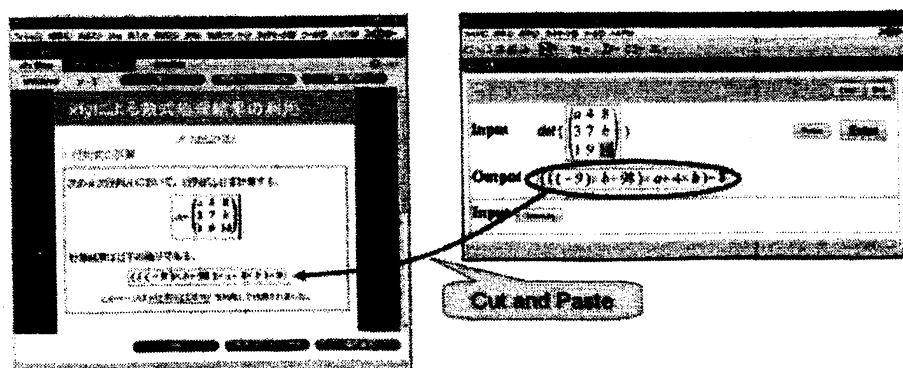


図 20: ブログ文書の編集

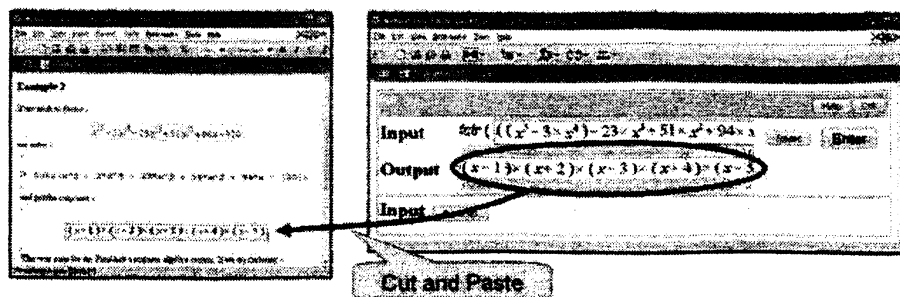


図 21: e-learning コンテンツの編集

6 おわりに

本研究では、数式の数学的意味を記述する MathML Content Markup を WYSIWYG で編集するプラグインの実装を行った。この xfy コンポーネントは、直接編集も可能なテンプレートエディタとして実装を行っており、数式エディタに必要な機能を備えるように注意した。数式処理機能の提供は従来から我々が行ってきた数学 Web サービスにより提供することで、xfy に数式処理機能を追加することができる。これにより、数式処理システムによる計算やその結果を文書編集にシームレスに利用できる数学文書編集環境が構築できる。

参 考 文 献

- [1] MathML, <http://www.w3.org/TR/MathML2/>
- [2] xfy, <https://www.xfytec.com/>
- [3] Formulator MathML Weaver, <http://mmlsoft.com/projects/formulator/>
- [4] Integre MathML Equation Editor, <http://www.integretechpub.com/zed/>
- [5] Luca Padovani and Ricardo Solmi: An Investigation on the Dynamics of Direct-Manipulation Editors for Mathematics, MKM 2004, Lecture Notes in Computer Science Vol.3119, pp.302-316, 2004
- [6] Kai Li, Masato Sakai, Yukihiro Morizane, Masahiro Kono, Matu-Tarow Noda, Lupin : Towards the framework of web-based problem solving environments, In *Proc. ATCM2003*, pages 276-285, 2003.
- [7] MONET Home Page, <http://monet.nag.co.uk>.